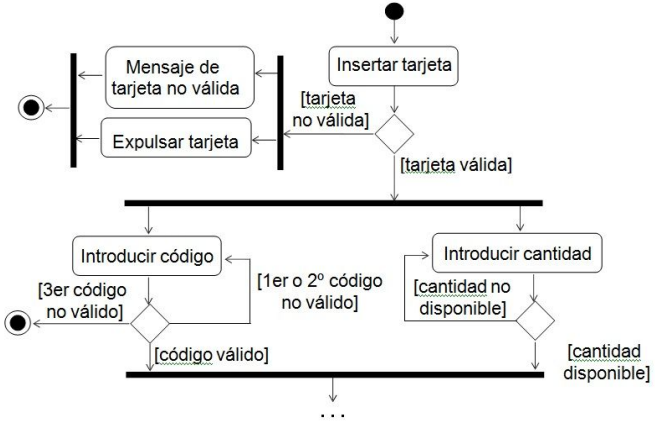
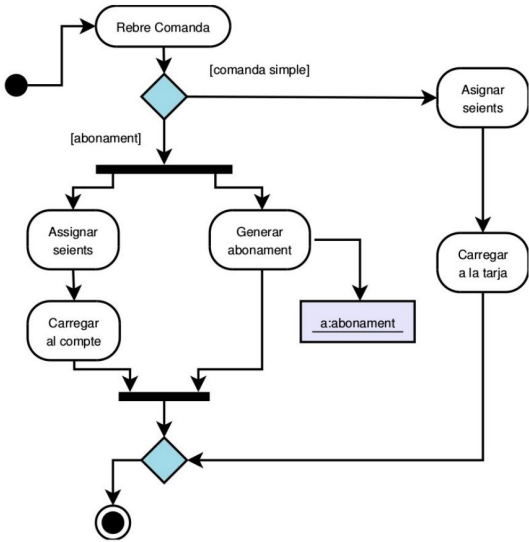
UML

Unified Modeling Lenguage

Ángel González M.



Diagramas de actividad (Activity diagrams)



Los diagramas de actividad

Los casos de uso muestran **que** debería hacer tu sistema

En algunas situaciones un caso de uso es más que suficiente para explicarlo todo.

Pero en otras ocasiones la lógica del problema puede ser muy complicada, y no es suficiente con representarla en un caso de uso.

Por tanto, muchas de las aplicaciones con las que nos tocará lidiar son complejas y requieren una análisis significativo.

Además de abordar la complejidad, necesitaremos alguna forma de testear la exactitud y el completado correcto del caso de uso.

Los diagramas de actividad **permiten describir como un sistema implementa su funcionalidad**.

Los diagramas de actividad **modelan el comportamiento dinámico de un procedimiento**, transacción o caso de uso haciendo énfasis en el proceso que se lleva a cabo.

Diagramas de actividad VS diagramas de flujo

Los diagramas de actividad **son parecidos a los diagramas de flujo** (que ya hemos visto en la 1ª eval), pero mucho más potentes. (no se quedan en flujos, condicionales y loops)

- NOTA: Los diagramas de flujo no son parte de UML
- Los **diagramas de actividad y de estado** si son parte de UML

Son útiles para describir procesos

No confundir entre diagrama de estado y diagrama de actividad

Los **Diagramas de Actividad** son muy parecidos a los DF (diagramas de Flujo) pero la diferencia radica en que todas **las actividades están unidas de forma clara a objetos.**
Los **Diagramas de Estado** muestran los diferentes "estados" de un objeto durante su ciclo de vida como también las acciones que generan estos cambios en el objeto.

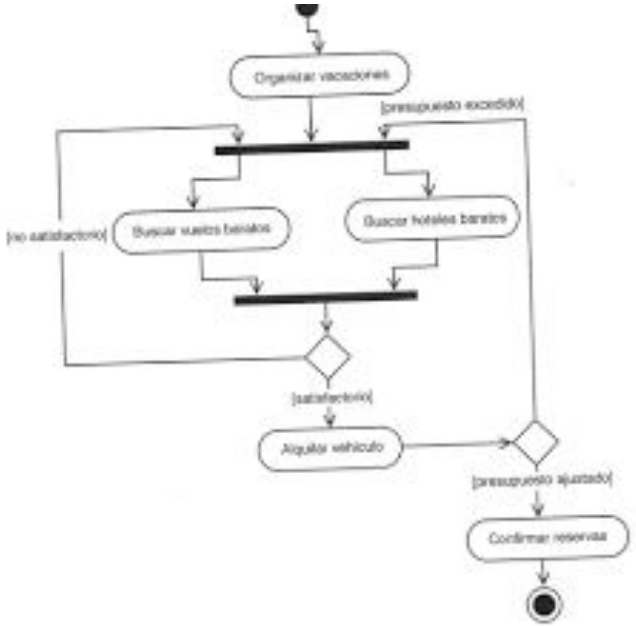
Por tanto un diagrama de actividad es una variación el diagrama de estado, donde cada "estado" representa a las distintas operaciones y cada transiciones representa las diferentes actividades que ocurren cuando la operación es completa.

Ángel González M.

Aplicación de los diagramas de actividad

Los diagramas de actividad son aplicables a cualquier tipo de modelado

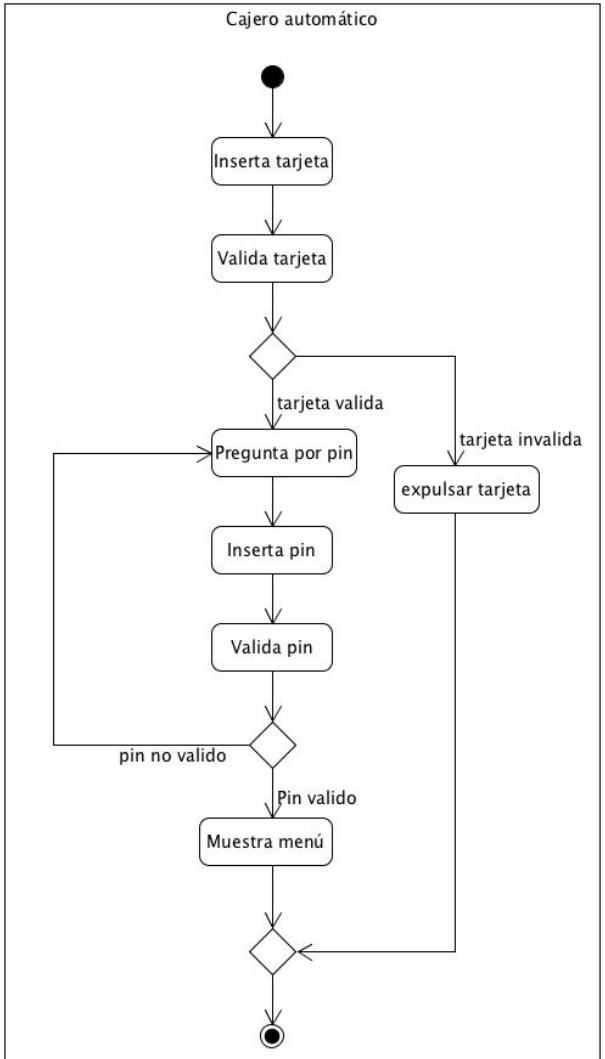
- procesos de negocio
- procesos de software
- workflows
- operaciones complejas
- reglas complejas de negocio
- casos de uso sencillos
- cualquier tipo de caso de uso



Cuando se utiliza para modelado de software, las actividades típicamente representan un comportamiento invocado como resultado de una llamada de método

Cuando se usa para modelado de negocio, las actividades podrían ser triggered(lanzadas) por eventos externos o eventos internos.

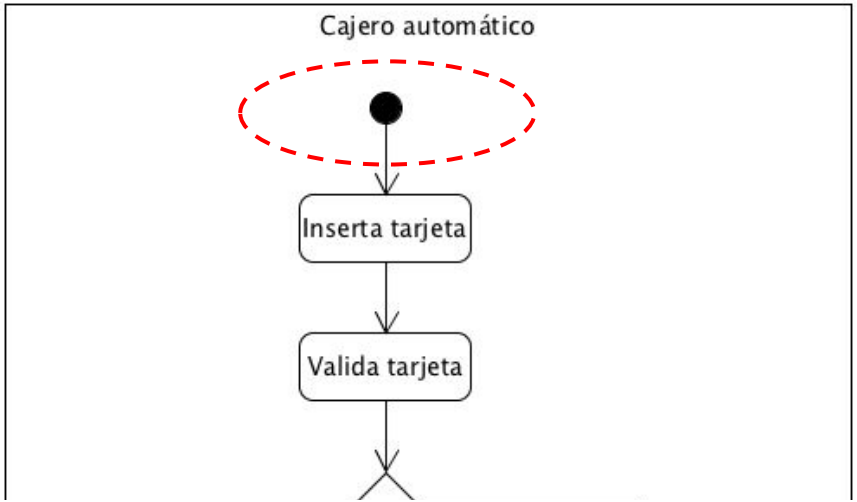
Ejemplo



Ángel González M.

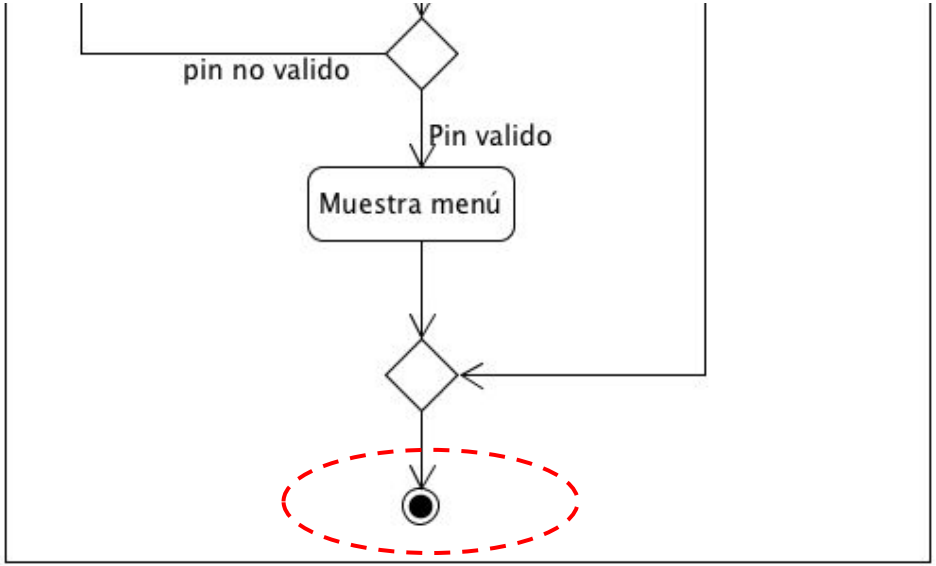
Partes básica de un diagrama de actividad

nodo inicial: simplemente marca el inicio de la actividad



Partes básica de un diagrama de actividad

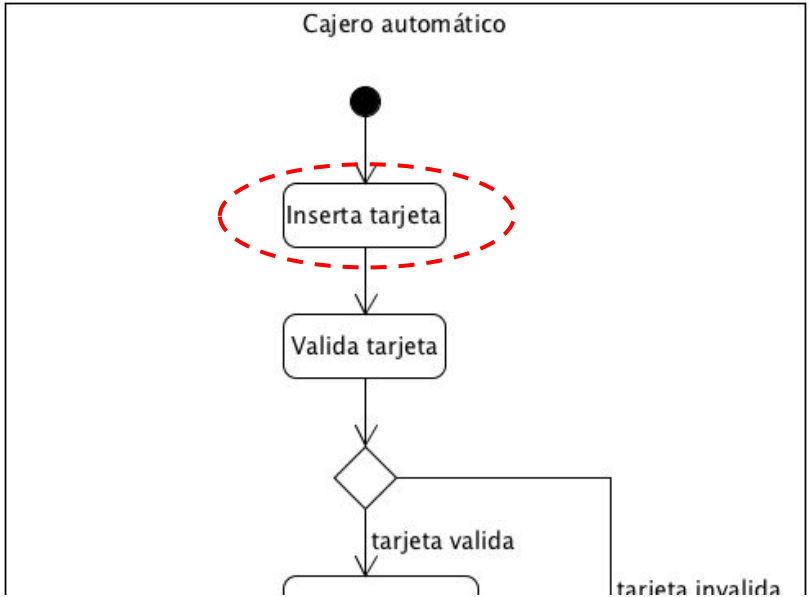
nodo final: simplemente marca el final de la actividad . Se representa con un círculo negro inscrito dentro de otro.



Partes básica de un diagrama de actividad

Las **acciones** son dibujadas como rectángulos.

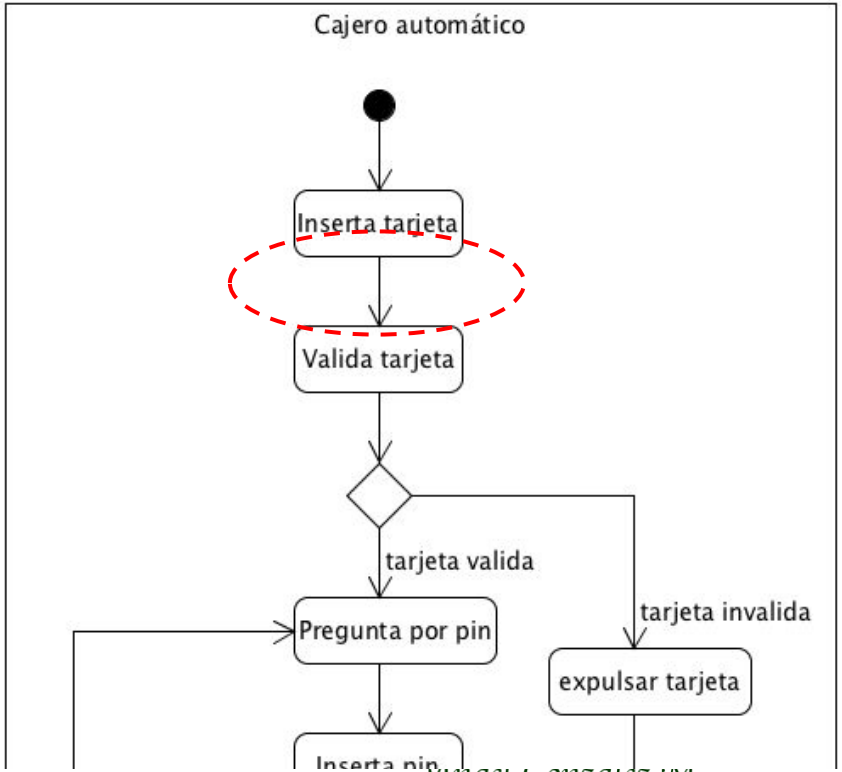
Las acciones son los pasos que tienen lugar en la actividad general: ej introduce tarjeta, válida tarjeta, etc



Partes básica de un diagrama de actividad

El flujo de la actividad es mostrado usando líneas con flechas llamadas caminos o paths

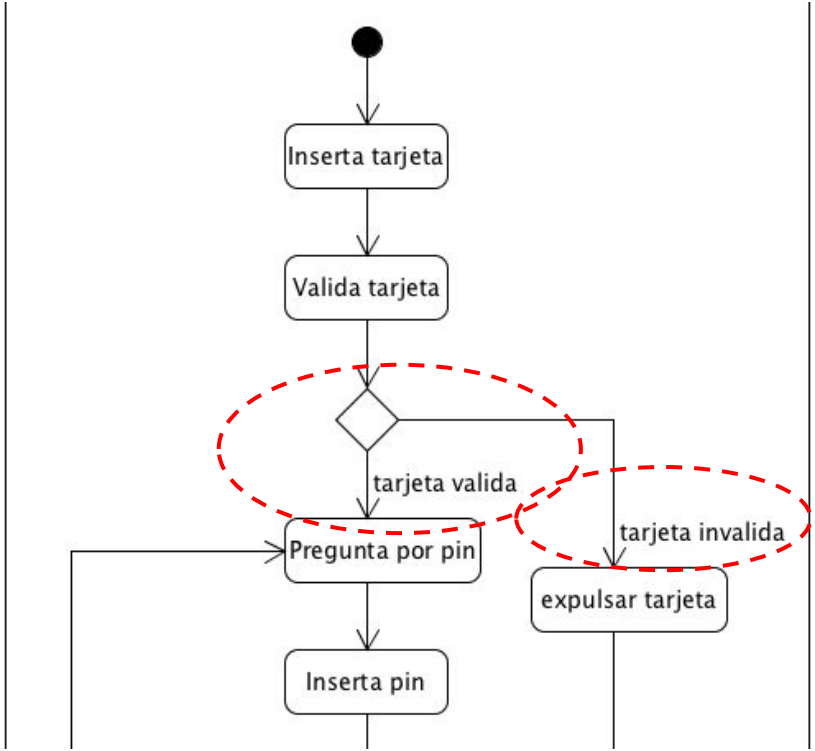
La flecha **muestra la dirección del flujo**.



Partes básica de un diagrama de actividad

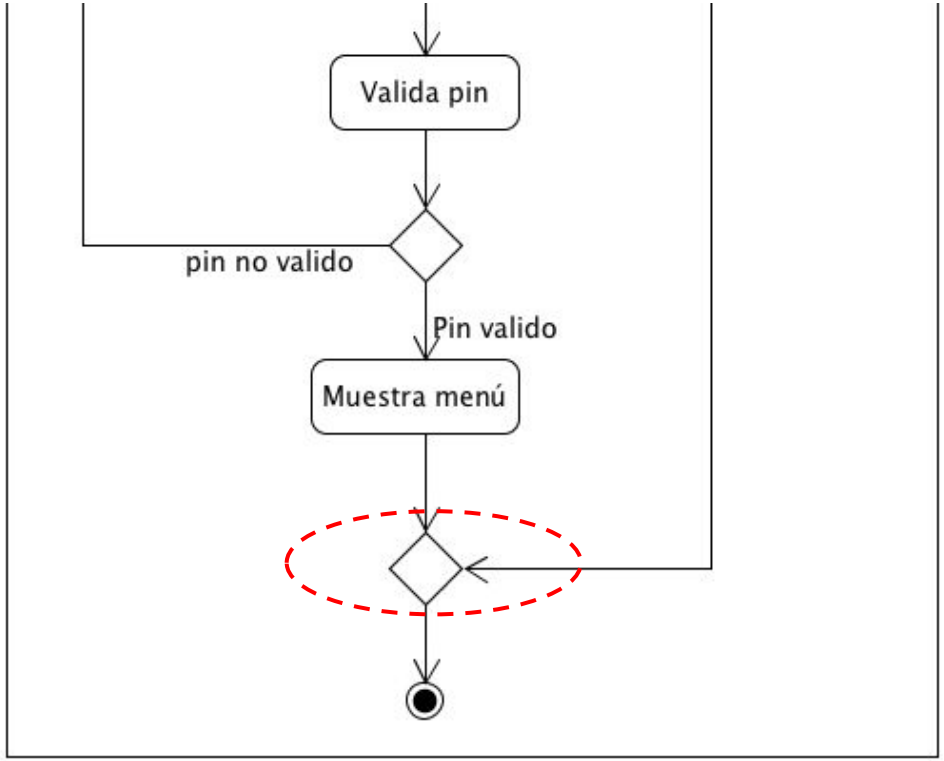
El primer rombo es llamado decisión (es similar al if-else)

Date cuenta de que las flechas que salen del rombo, están etiquetadas con las condiciones que deberían ser ciertas para que el flujo continúe por ese path (ruta)



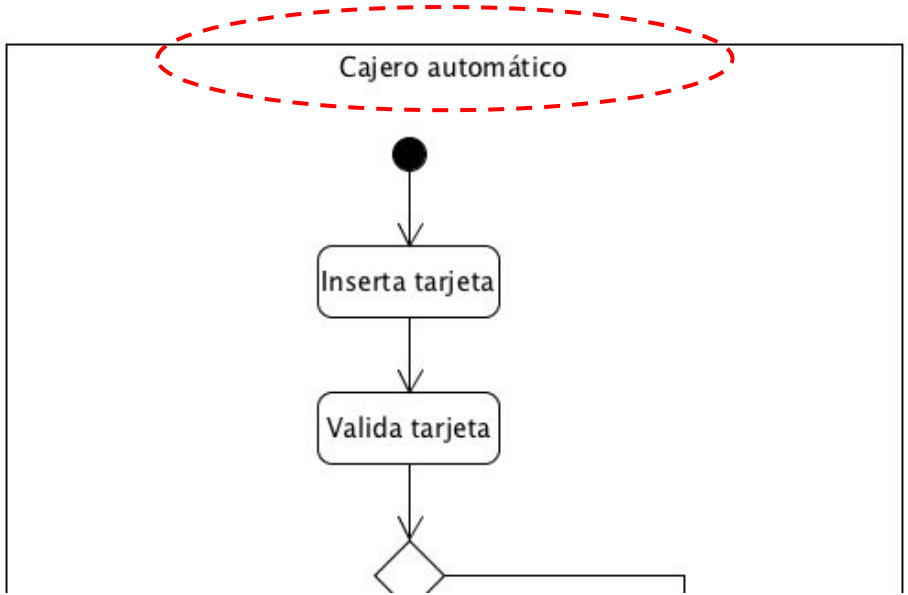
Partes básica de un diagrama de actividad

El rombo del final es llamado merge(mezcla). Sirve para unir diferentes paths (rutas)



Partes básica de un diagrama de actividad

Al conjunto del diagrama se le puede etiquetar con un texto en la parte superior



Notación en los diagramas de actividad

Ángel González M.

Acción

A un **simple trozo de comportamiento** se le llama acción.

Una acción podría ser:

- get/set el valor de un atributo
- invocar un método o función de otra clase
- llamada a una función
- invocar una actividad que contiene acciones
- enviar una señal o notificación de un evento a un grupo de objetos

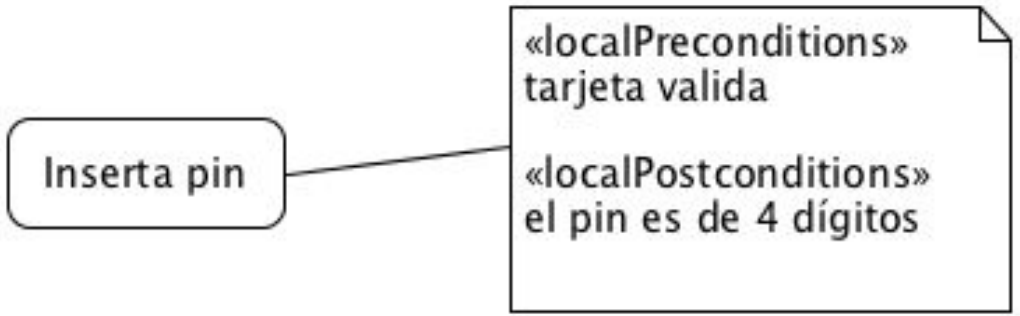
Acciones con precondiciones y postcondiciones

Para acciones más complejas, se puede especificar las **condiciones que deben cumplirse antes y después** de ejecutar la acción.

Para ello, usamos una nota enganchada a la acción, **anotándola** con el **estereotipo**.

<<localPrecondition>>

<<localPostcondition>>



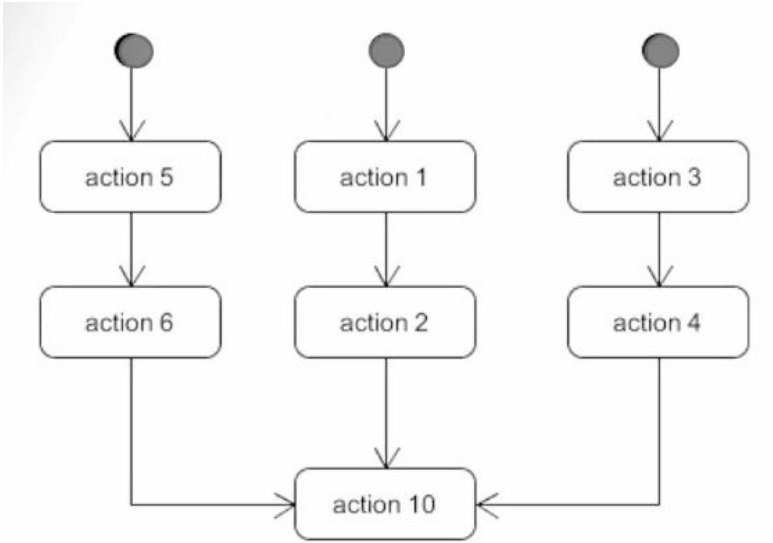
Explicación de acciones

Puedes explicar **acciones más complicadas**, escribiendo **pseudocódigo** en el interior de la acción si fuese necesario.

Nodo inicial (initial node)

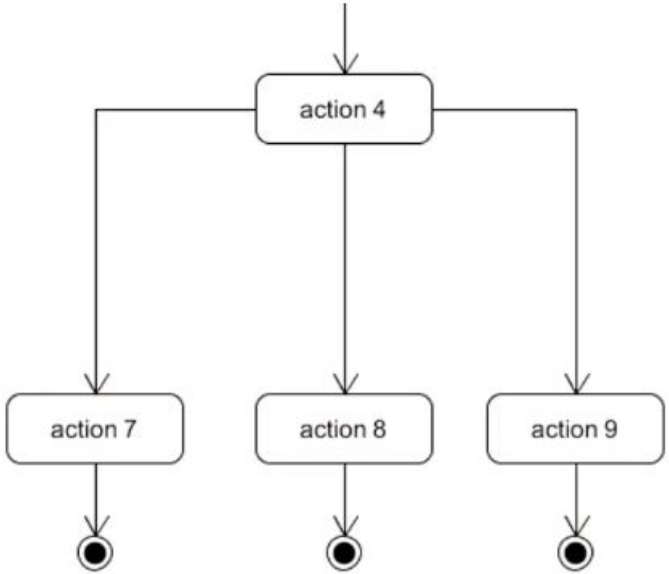
Marca el punto de inicio de la actividad

El diagrama de actividad puede tener varios puntos de inicio



Nodo final (final node)

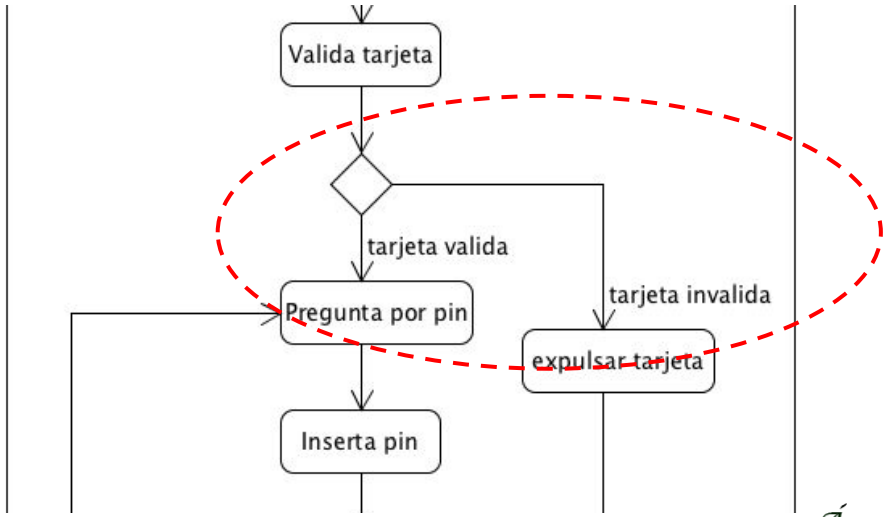
Determina el final del diagrama de actividad



Ángel González M.

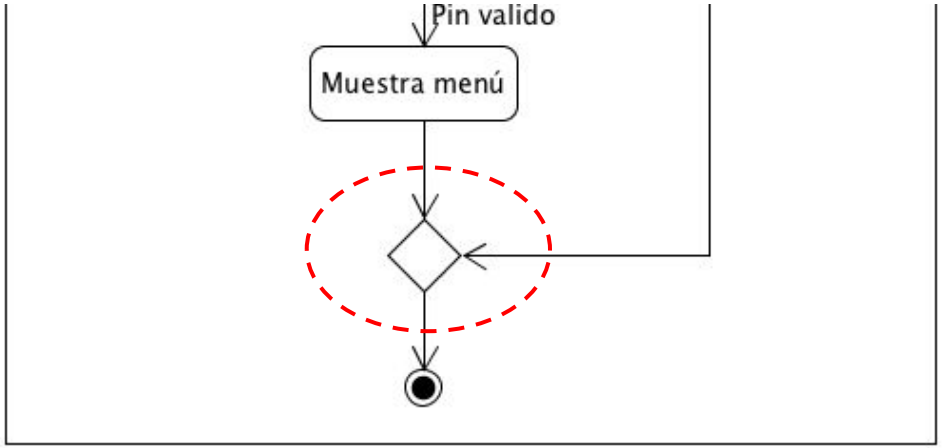
Decisión (decisión)

Utiliza una condición para asegurarse de que un objeto o flujo de control desciende sólo por un camino



Mezcla/fundir (Merge)

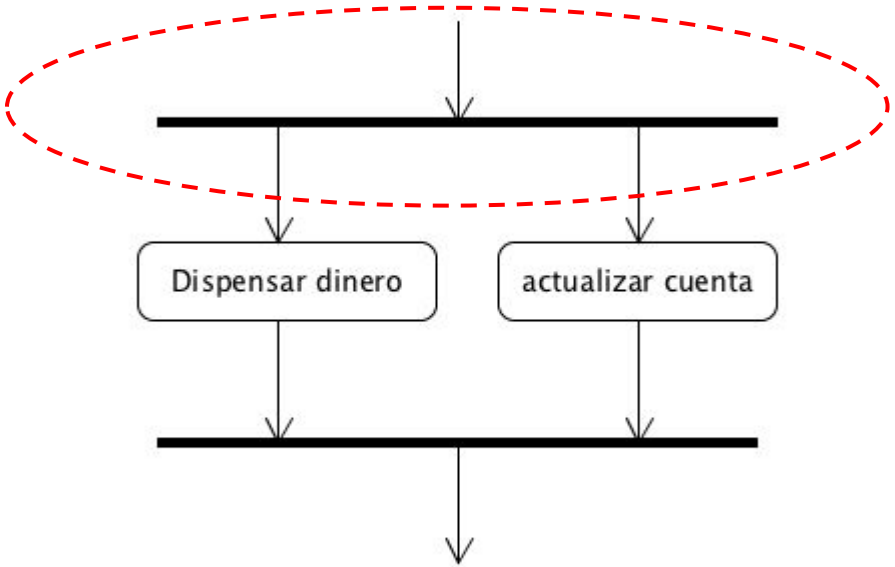
Une varios caminos



Bifurcación (fork)

En ocasiones necesitas que actividades o acciones **ocurran en paralelo**.

Para dividir el comportamiento en operaciones concurrentes(a la vez) usa el **nodo fork.**



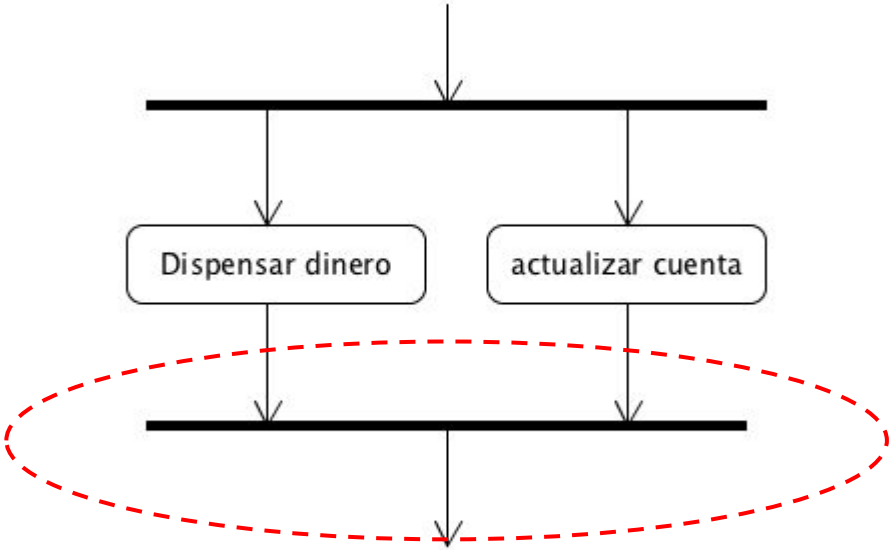
Unión (Join)

Sincroniza múltiples flujos de la actividad en un flujo único de ejecución

En este caso, se ejecutará concurrentemente las acciones de dispensar dinero y actualizar cuenta.

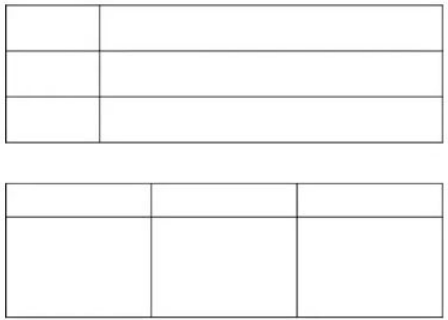
Puede que una acción lleve más tiempo que la otra en ejecutarse.

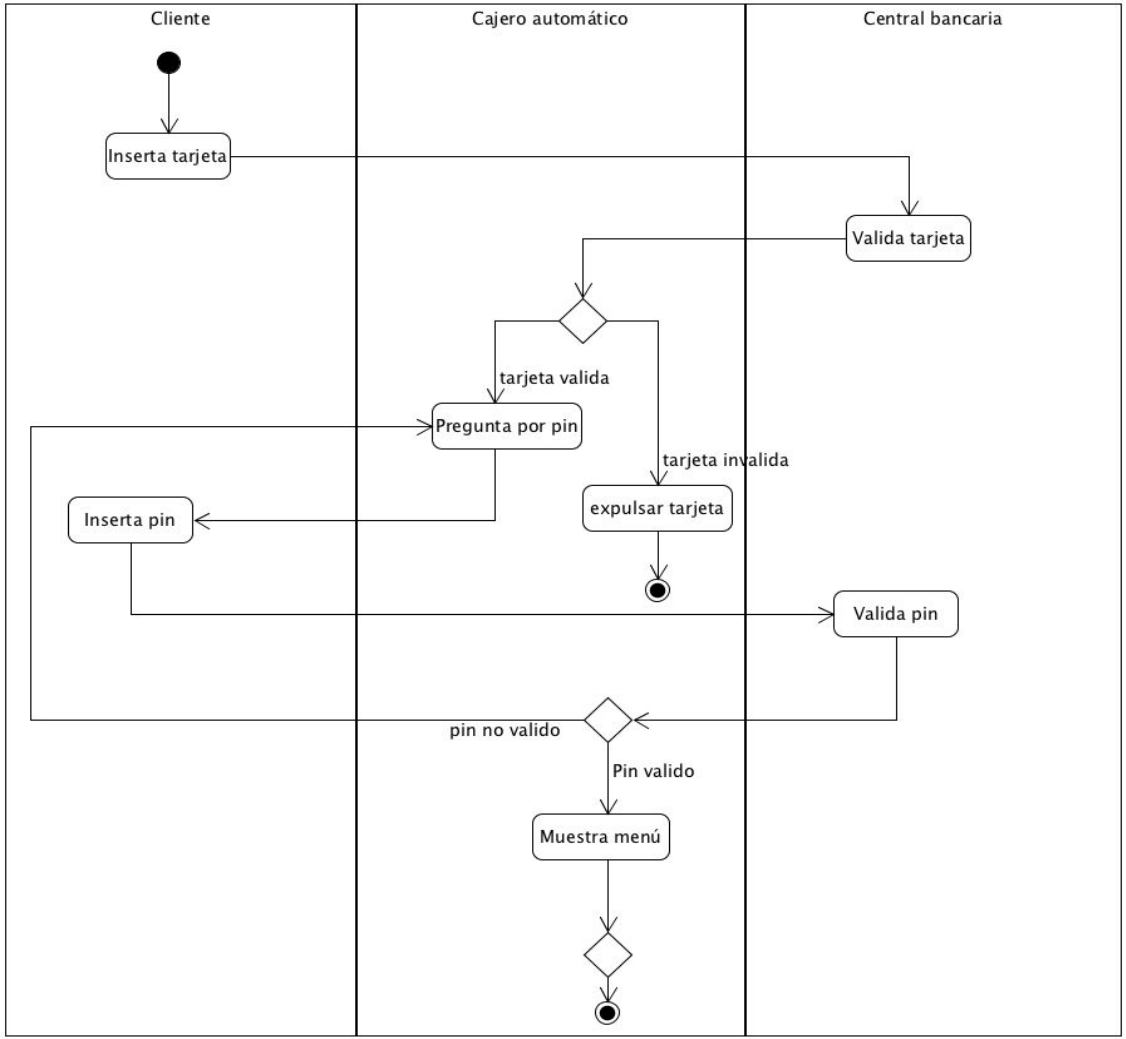
Solamente cuando las 2 tareas hayan concluido se unirán(join) y se continuará con el flujo normal del diagrama de actividad.



Particiones (Swim lanes)

Son útiles para indicar que hace cada una de las acciones en el diagrama de actividad.

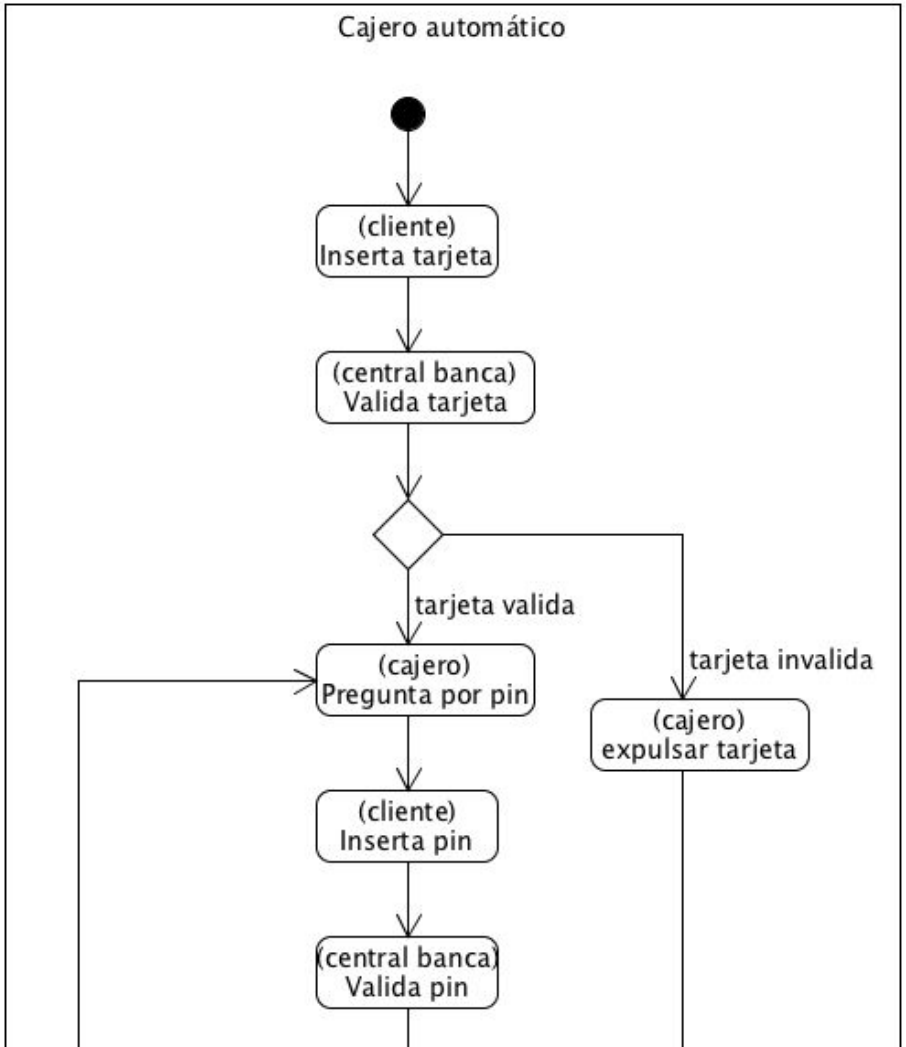




Particiones (Swim lanes)

También podemos expresar las swim lanes (particiones) de esta forma.

Anteponiendo al nombre de la actividad, entre paréntesis, el nombre del actor



Enviando y recibiendo señales

Nuestro software interacciona con personas, sistemas o procesos externos.

Las señales representan interacciones con participantes externos.

Ejemplos de señales son:

- El software envía una petición a la banca central para validar la tarjeta
- La pulsación de una tecla del ratón
- El sistema notifica al cliente que su pedido ha sido enviado.
- Se envía o recibe un mensaje del servidor
- Cada cierto tiempo hay que realizar una acción
- ...

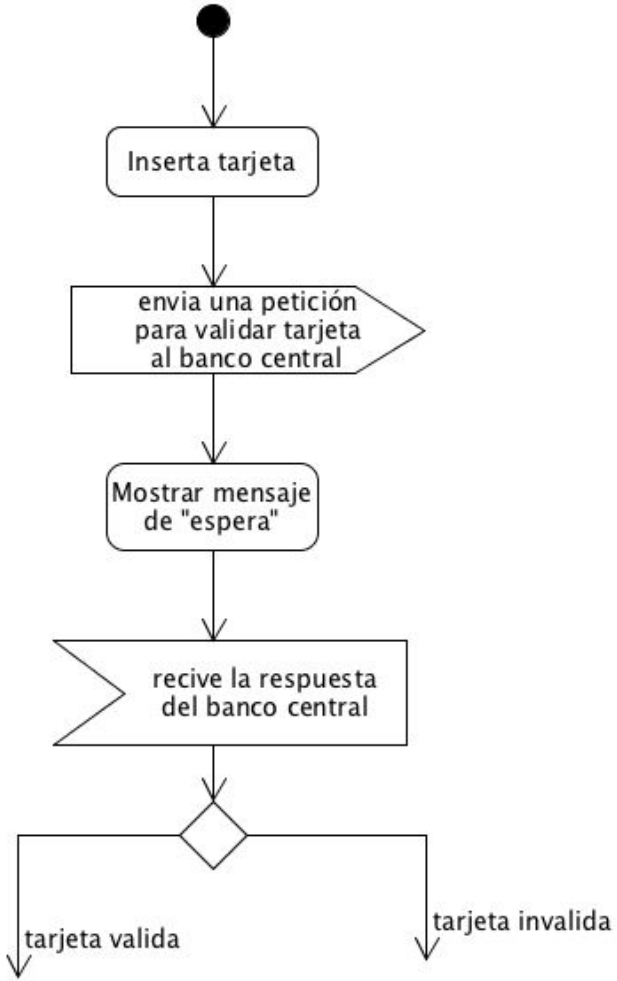
señal de recepción(receive signal)

Tiene el efecto de iniciar una acción en tu diagrama de actividad

El sistema se queda esperando la recepción de una señal. Este espera que la señal llegue alguna vez pero no sabe exactamente cuando.

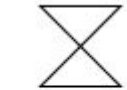
Enviar señal (send signal)

Son señales que se envían a un participante externo.



Ángel González M.

Eventos de tiempo (time event)



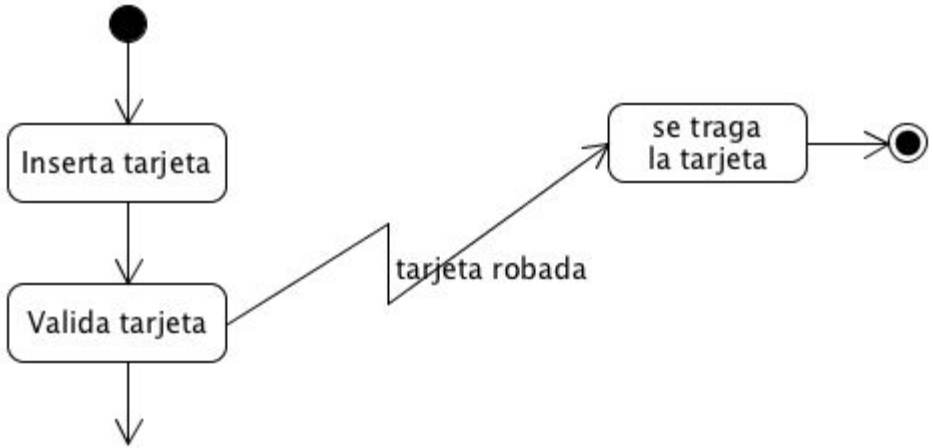
En este caso, el evento de tiempo espera 2 segundos antes de refrescar la barra de progreso.

Si no va precedida de ninguna acción implica que el evento de tiempo ocurre cada 3 minutos

Manejo de excepciones (exception handler)

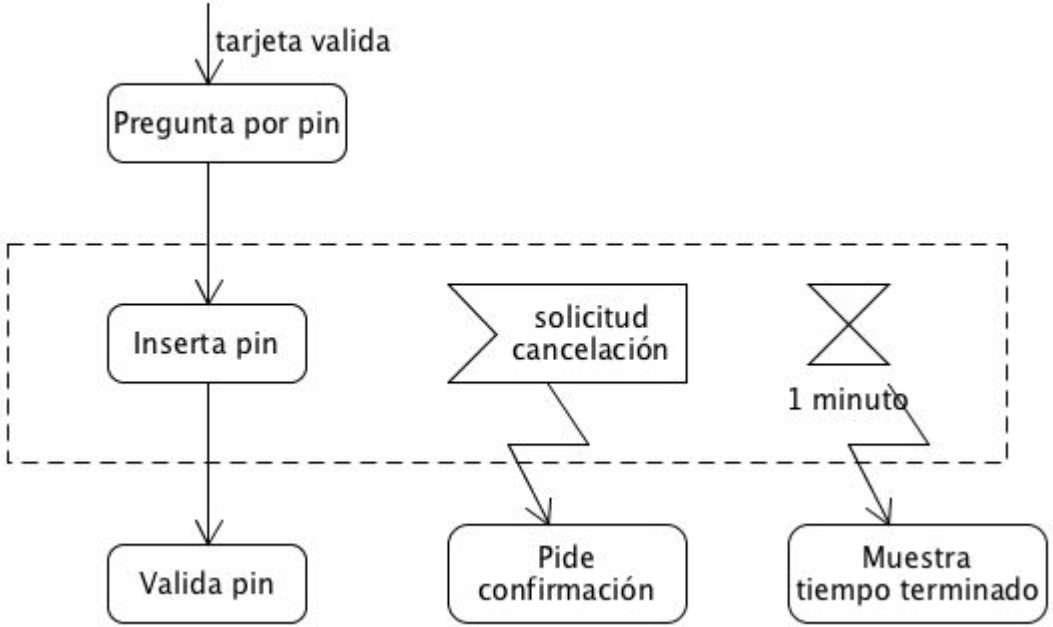
Una excepción es un evento que ocurre durante la ejecución del programa y que **hace que el flujo normal de ejecución se rompa**.

Un manejador de excepciones define el tipo de excepción que ha ocurrido y lo que se va a ejecutar cuando la excepción sea capturada.



Regiones de actividad Interruptible
Interruptible Activity Regions

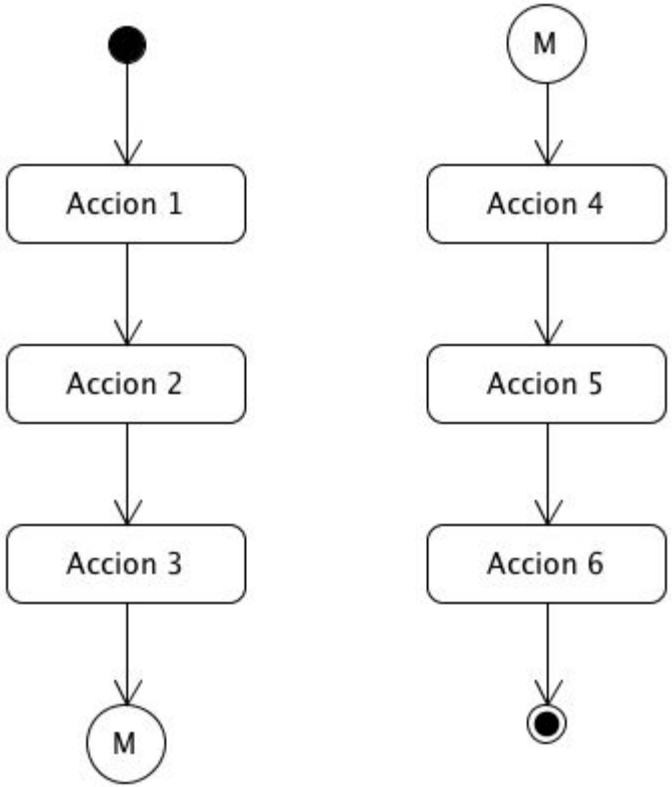
La **región** de **actividad interrumpible** rodea un grupo de elementos de actividad, todos afectados por ciertas interrupciones de tal manera que todos los tokens que pasan dentro de la región se terminan en caso de que se disparen las interrupciones.



Conectores

Son saltos en el diagrama.

Nos ayudan a presentar de forma más simple la información, permitiendo que un diagrama continúe en otro lugar

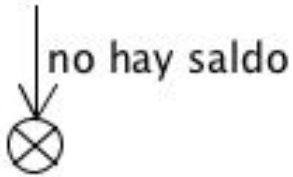


Ángel González M.

Nodo Final de flujo

Indica que el flujo termina ahí.

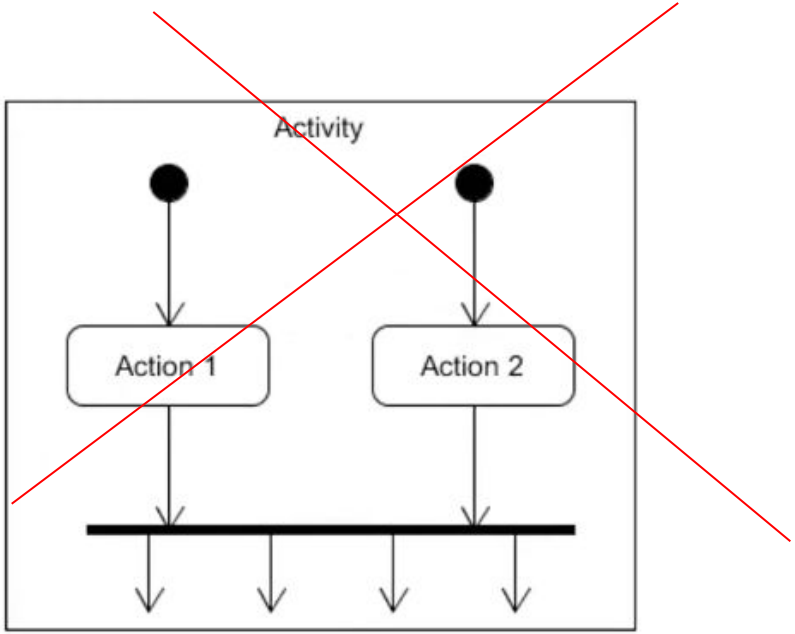
No tiene efecto sobre otros flujos del diagrama

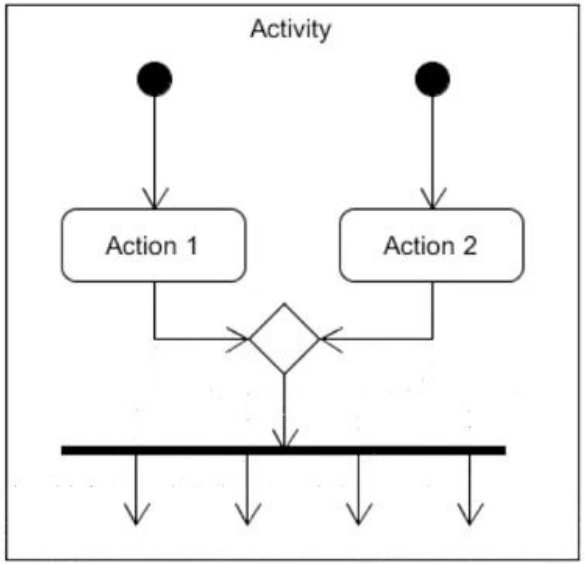


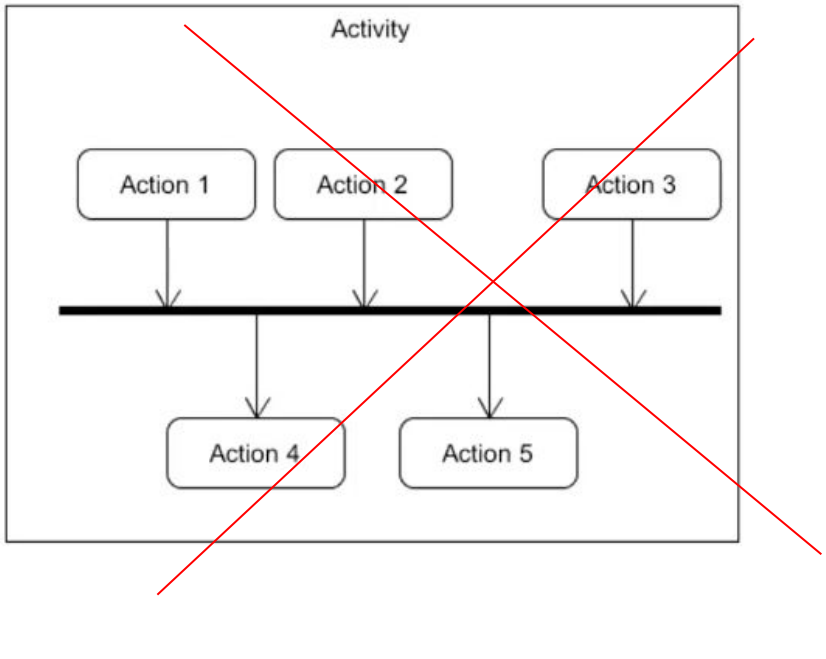
Buenas prácticas para hacer diagramas de actividad

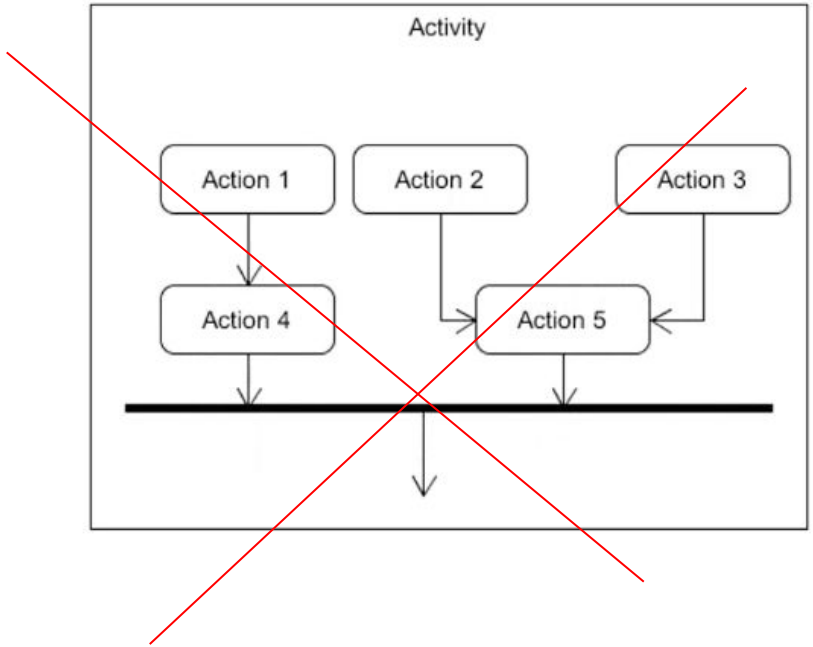
- Pon el nodo inicial en la parte superior izquierda
- Añade un nodo final para hacer más entendible el diagrama.
- No crees actividades sin una línea de entrada
- No crees actividades sin una línea de salida
- En los nodos de decisión, pon siempre una etiqueta indicando su condición
- En los nodos de decisión no dejes condiciones sin cubrir. Ej cantidad>0 cantidad<0 (faltaría cantidad=0)
- En los nodos de decisión no solapes condiciones. Ej cantidad<=0 cantidad>=0
- Si usas swim lanes no uses más de 5 columnas

Ángel González M.


Activity
Action 1
Action 2

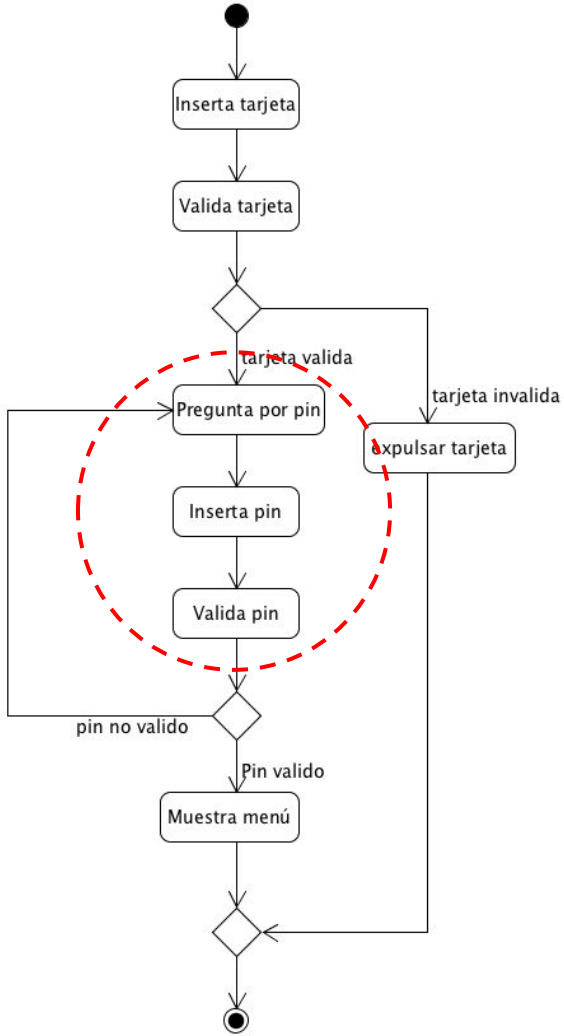

Activity
Action 1
Action 2

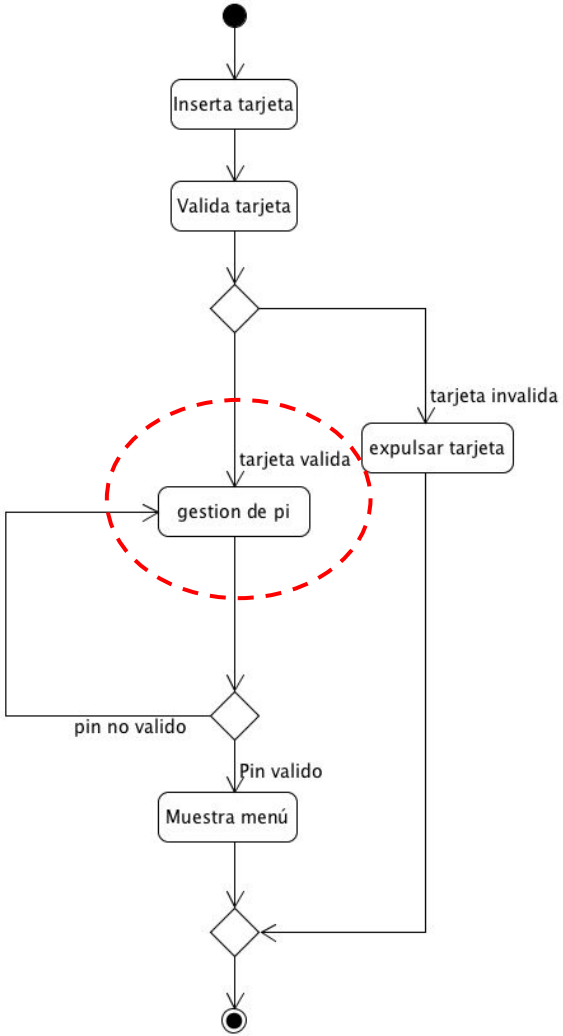




Ángel González M.

Si el diagrama te está quedando muy grande, organízalo en actividades de más alto nivel





Ejercicio

Puedes entender este diagrama?